Research Areas

Stephen A. Edwards

Department of Computer Science,
Columbia University

www.cs.columbia.edu/~sedwards

sedwards@cs.columbia.edu

Embedded Systems

Computers masquerading as something else.

Casio
Camera
Watch

Nokia 7110
Browser
Phone

Sony
Playstation 2

Philips
DVD Player

Philips
TiVo Recorder

Long-Term Goal

Supplying tools that speed the development of embedded systems.

Domain-Specific Languages

Little languages that fit the problem

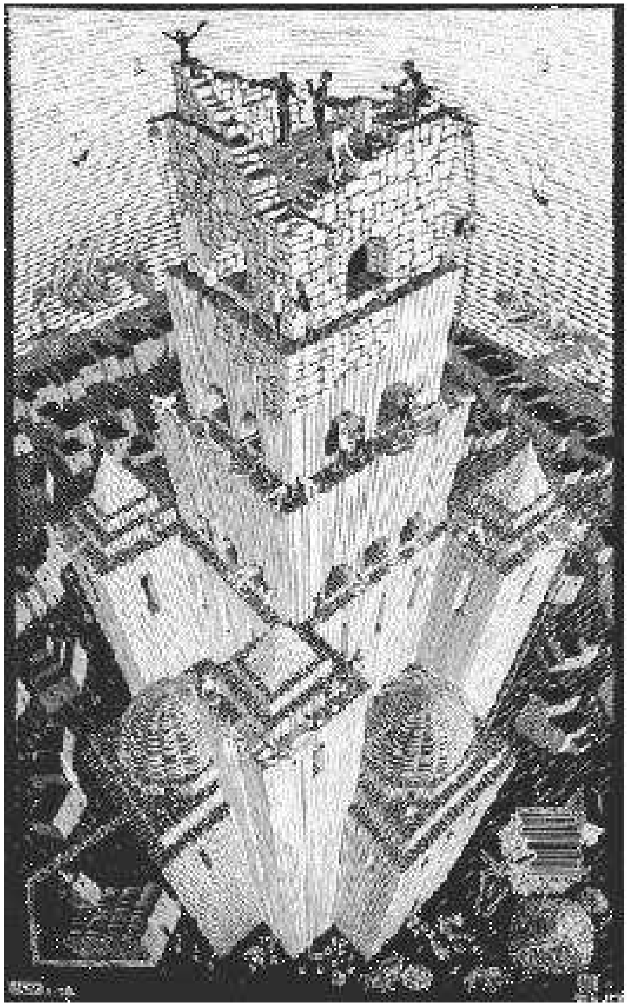
More succinct description that are

1. Quicker to create
2. Easier to get right

More opportunities for optimization

General-purpose languages hindered by undecidability

Domain-specific languages much simpler



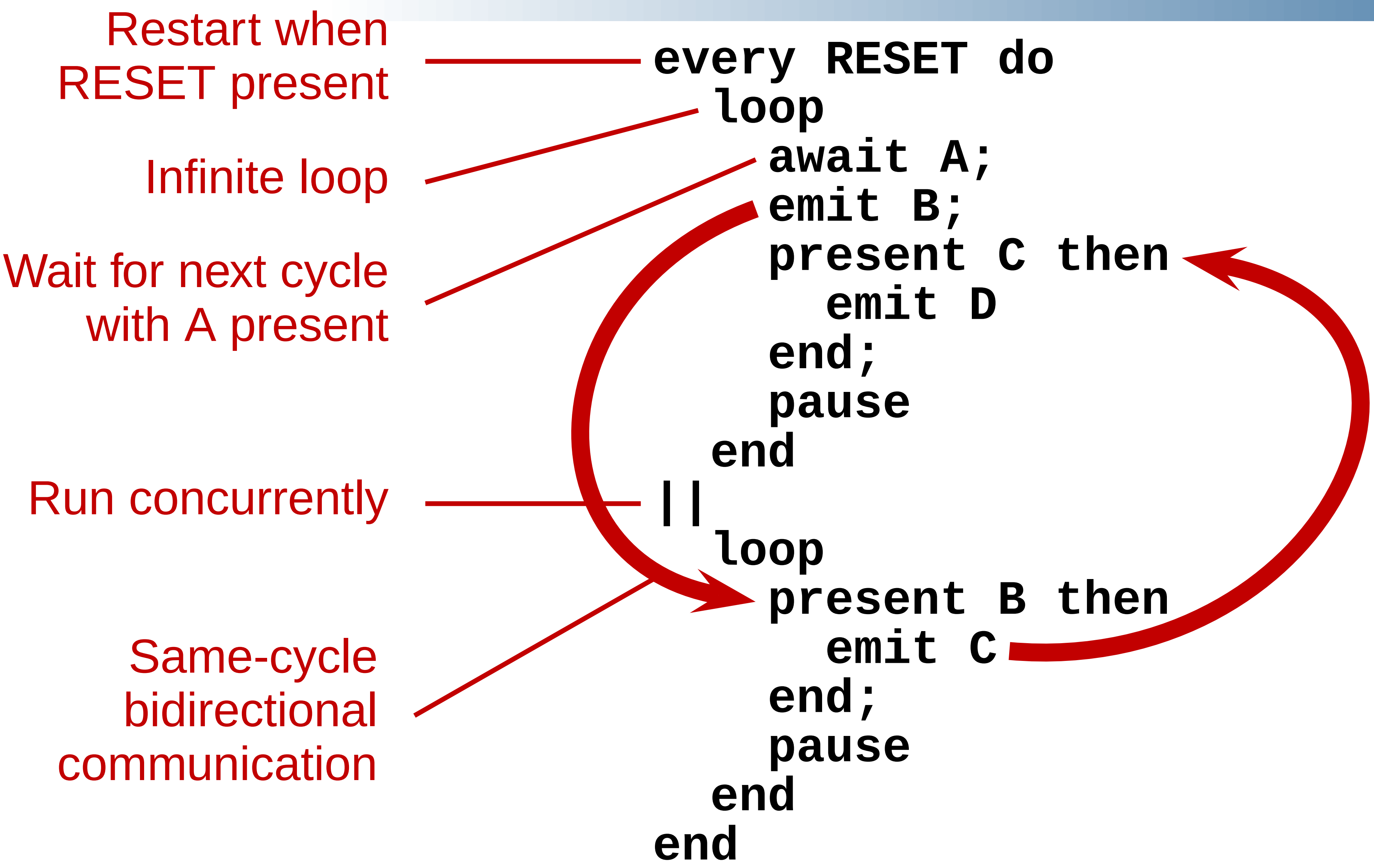
Real-time Languages: Esterel

Synchronous language developed by Gérard Berry in France

Basic idea: use global clock for synchronization in software

Challenge: How to combine concurrency, synchronization, and instantaneous communication

Esterel



Restart when RESET present

Infinite loop

Wait for next cycle with A present

Run concurrently

Same-cycle bidirectional communication

```
every RESET do
  loop
    await A;
    emit B;
    present C then
      emit D
    end;
    pause
  end
||
  loop
    present B then
      emit C
    end;
    pause
  end
end
```

Esterel

Previous work:

- Compiler that speed up certain large programs 100×
- Has limitations (e.g., owned by former employer)

Current projects

- New compiler infrastructure designed for research
- Better circuits from Esterel programs (Jia Zheng)
- Faster code from PDGs (Cristian Soviani)
- An Esterel Virtual Machine interpreter for small-footprint applications (Aruchunan Vaseekaran and Tamara Blain)

Languages for Device Drivers

Device drivers are those pieces of software that you absolutely need that never seem to work

Big security/reliability hole: run in Kernel mode

Responsible for 80% of all Windows crashes

Tedious, difficult-to-write

Ever more important as customized hardware proliferates

Best-to-date

Thibault, Marlet, and Consel

IEEE Transactions Software Engineering, 1999

Developed the Graphics Adaptor Language for writing XFree86 video card drivers

Report GAL drivers are 1/9th the size of their C counterparts

No performance penalty

GAL S3 Driver (fragment)

```
chipsets S3_911, S3_924;
```

What driver supports

```
port svga index := 0x3d4;
port misc := 0x3cc, 0x3c2;
```

Write address, then data

```
register ChipID := sva(0x30);
```

Logical register

```
serial begin
  misc[3..2] <= (3,- , -, -, -) W;
  seq(0x12) <=> (-, PLL1, -, -, -) R/W;
end;
```

Access sequence for register

```
identification begin
1: ChipID[7..4] =>
   (0x8 => step 2, 0x9 => S3_928);
2: ChipID[1..0] =>
   (0x1 => S3_911, 0x2 => S3_924);
```

Rules for identifying card

Ongoing Work

Develop language for network card drivers under Linux (Chris Conway)

Study many existing implementations (Noel Vega)

Develop prototype language, compiler

Explore challenge of porting to other OSes

Apply lessons to other classes of drivers

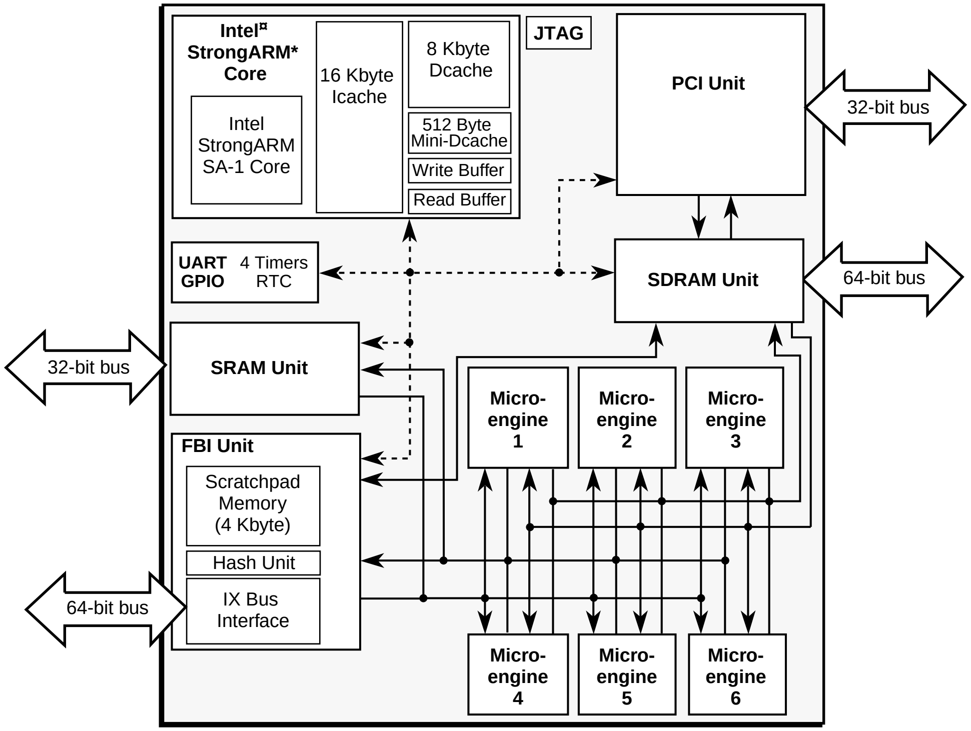
Domain-specific Languages for Emerging Architectures

The sovereignty of the general-purpose processor is ending.

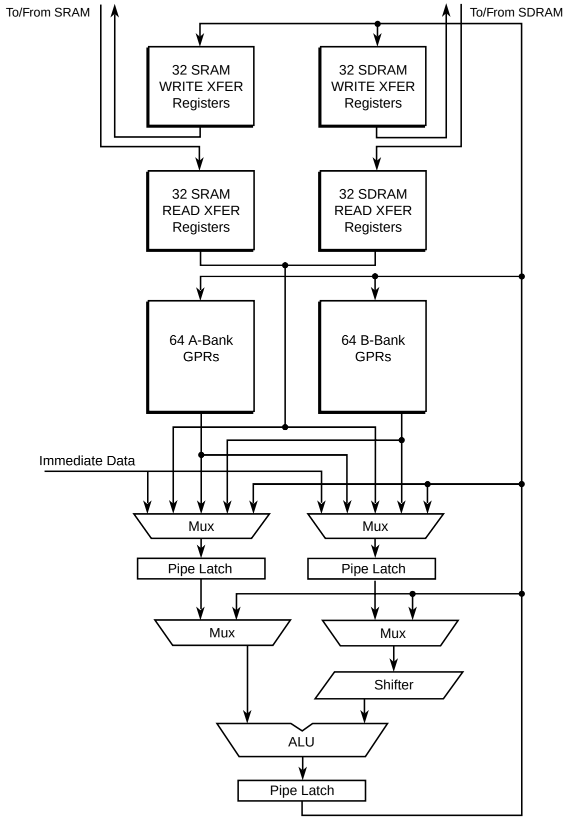
Silicon is getting so cheap, we can “waste” it in special-purpose applications:

- Digital Signal Processors
- Graphics pipelines in videogames
- Network Processors

Intel’s IXP1200 Network Processor



Intel IXP1200 Network Processor



Really powerful, but nobody can program it.

StrongARM + 6 concurrent microengines

How to program these architectures?

Most now programmed in assembly language.

Not practical for ever-growing system complexity

C isn’t going to cut it: these are not PDP-11s

We need new languages and compilers to go with them

Domain-specific languages and compilers

Project just starting (with Al Aho)

Goal is to look at a variety of emerging architectures, propose new languages for them, and devise optimizing compilation algorithms

We hope to do for these different architectures what FORTRAN did for general-purpose computers

Interested? Pester us.

Thank you